

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 11-175327

(43)Date of publication of application : 02.07.1999

(51)Int.Cl.

G06F 9/06

G06F 9/06

G06F 9/44

(21)Application number : 09-341654

(71)Applicant : CLASS TECHNOLOGY:KK

(22)Date of filing : 11.12.1997

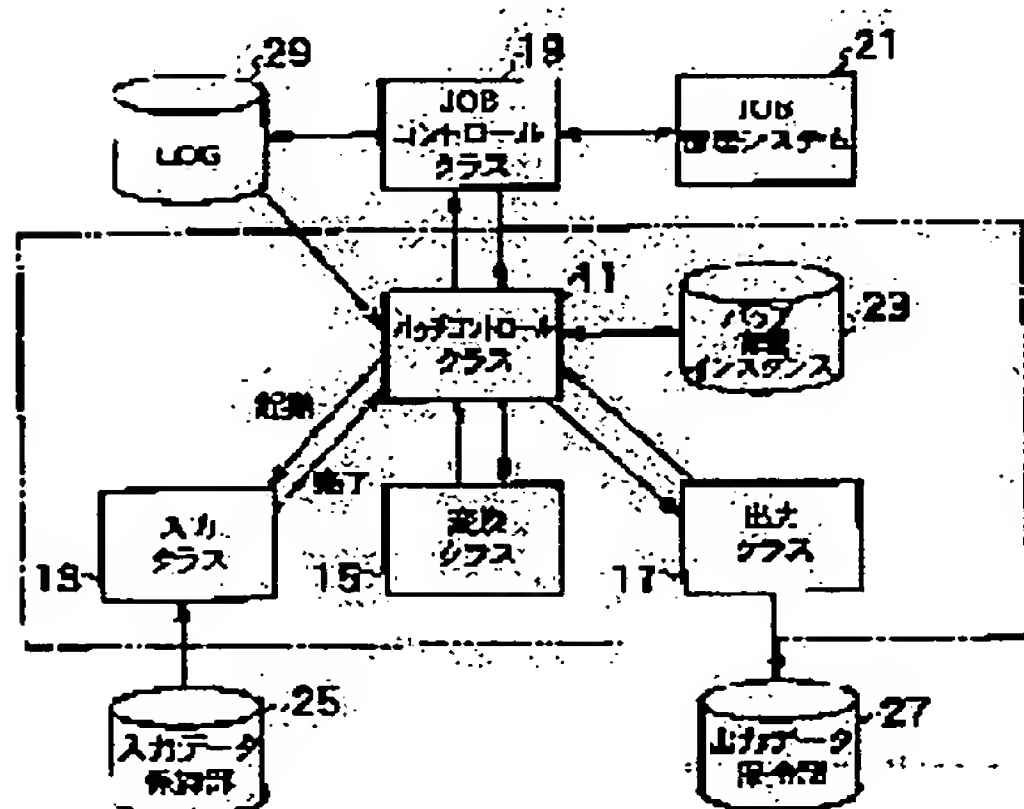
(72)Inventor : SHIKURA MIKIO

(54) INFORMATION PROCESSOR, INFORMATION PROCESSING METHOD AND
COMPUTER READABLE RECORDING MEDIUM RECORDING INFORMATION
PROCESSING PROGRAM

(57)Abstract:

PROBLEM TO BE SOLVED: To easily change the contents of the procedure of a processing or the like and to process information without the need of programming.

SOLUTION: This information processor is provided with an input class 13 for performing the input processing of data by prescribed conditions in the case of receiving a start message from a batch control class 11 and outputting an end message to the batch control class 11, a conversion class 15 for converting the data whose input processing is performed by the prescribed conditions in the case of receiving the start message from the batch control class 11 and outputting the end message to the batch control class 11 and an output class 17 for performing an output processing for the conversion processed data by the prescribed conditions in the case of receiving the start message from the batch control class 11 and outputting



BEST AVAILABLE COPY

- the end message to the batch control class 11. The classes are controlled by the batch control class 11.

11.4.4.4
11.4.4.4.1

11.4.4.4.2
11.4.4.4.3

(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号

特開平11-175327

(43)公開日 平成11年(1999) 7 月 2 日

(51)Int.Cl.⁶

G 0 6 F 9/06

9/44

識別記号

5 3 0

4 1 0

5 3 0

F I

G 0 6 F 9/06

9/44

5 3 0 T

4 1 0 B

5 3 0 P

審査請求 未請求 請求項の数10 O L (全 14 頁)

(21)出願番号

特願平9-341654

(22)出願日

平成9年(1997)12月11日

(71)出願人 597008717

株式会社クラステクノロジー

東京都新宿区西新宿7-5-20

(72)発明者 四倉 幹夫

東京都新宿区西新宿7丁目5番20号 株式
会社クラステクノロジー内

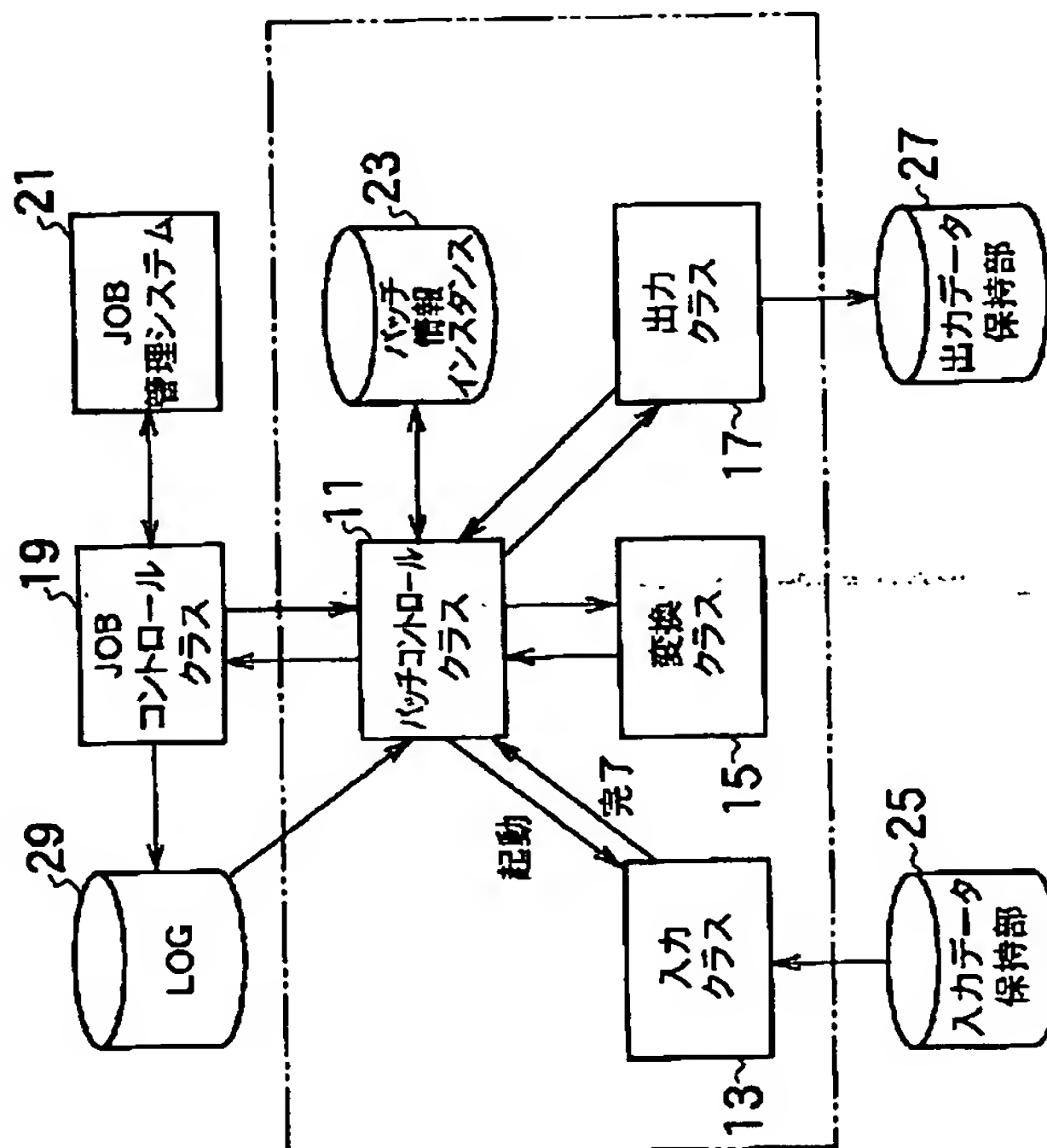
(74)代理人 弁理士 三好 秀和 (外8名)

(54)【発明の名称】 情報処理装置、情報処理方法及び情報処理プログラムを記録したコンピュータ読み取り可能な記録媒体

(57)【要約】

【課題】 処理の手続きの内容等の変更を容易に行い、さらに、プログラミングを必要とせずに情報処理を行うことである。

【解決手段】 バッチコントロールクラス11からの開始メッセージを受け取った場合に、所定の条件でデータの入力処理を行い、バッチコントロールクラス11へ終了メッセージを出力する入力クラス13と、バッチコントロールクラスからの開始メッセージを受け取った場合に、所定の条件で入力処理を行ったデータの変換を行い、バッチコントロールクラス11へ終了メッセージを出力する変換クラスと、バッチコントロールクラス11からの開始メッセージを受け取った場合に、所定の条件で変換処理データについて出力処理を行い、バッチコントロールクラス11へ終了メッセージを出力する出力クラス17と、を備え、これらのクラスをバッチコントロールクラス11が制御するようにしてある。



【特許請求の範囲】

【請求項 1】 オブジェクト指向技術を応用したクラスを用いて情報処理を行う装置であって、

開始メッセージをパッシングし、クラスからの終了メッセージの入力により次のクラスをスケジュールするバッチコントロールクラスと、

前記バッチコントロールクラスからの開始メッセージを受け取った場合に、所定の条件でデータの入力処理を行い、前記バッチコントロールクラスへ終了メッセージを出力する入力クラスと、

前記バッチコントロールクラスからの開始メッセージを受け取った場合に、所定の条件で前記入力処理を行ったデータの変換を行い、前記バッチコントロールクラスへ終了メッセージを出力する変換クラスと、

前記バッチコントロールクラスからの開始メッセージを受け取った場合に、所定の条件で前記変換処理データについて出力処理を行い、前記バッチコントロールクラスへ終了メッセージを出力する出力クラスと、
を備えることを特徴とする情報処理装置。

【請求項 2】 前記所定の条件は、

各クラスの処理内容を定義するパラメータにより特定されることを特徴とする請求項 1 記載の情報処理装置。

【請求項 3】 オブジェクト指向技術を応用したクラスを用いて情報処理を行うプログラムを作成する装置であって、

各クラスの処理内容を定義するパラメータを入力する入力部と、

入力した前記パラメータに基づいて、開始メッセージをパッシングし、クラスからの終了メッセージの入力により次のクラスをスケジュールするバッチコントロールクラス、前記バッチコントロールクラスからの開始メッセージを受け取った場合に、所定の条件でデータの入力処理を行い、前記バッチコントロールクラスへ終了メッセージを出力する入力クラス、前記バッチコントロールクラスからの開始メッセージを受け取った場合に、所定の条件で前記入力処理を行ったデータの変換を行い、前記バッチコントロールクラスへ終了メッセージを出力する変換クラス、及び、前記バッチコントロールクラスからの開始メッセージを受け取った場合に、所定の条件で前記変換処理データについて出力処理を行い、前記バッチコントロールクラスへ終了メッセージを出力する出力クラス、を実行するプログラムを生成するプログラム生成部と、
を備えることを特徴とする情報処理装置。

【請求項 4】 前記入力部は、

ウィザード画面を用いて入力を受け付けることを特徴とする請求項 3 記載の情報処理装置。

【請求項 5】 オブジェクト指向技術を応用したクラスを用いて情報処理を行う方法であって、

第一の開始メッセージをパッシングする第一のパッシン

グステップと、

前記第一の開始メッセージを受け取った場合に、データの入力を行って入力処理データを作成し、入力処理が完了した場合には、第一の終了メッセージを出力する入力処理ステップと、

前記第一の終了メッセージを入力した場合には、第二の開始メッセージをパッシングする第二のパッシングステップと、

前記第二の開始メッセージを受け取った場合に、前記入力データの変換を行って変換処理データを作成し、変換処理が完了した場合には、第二の終了メッセージを出力する変換処理ステップと、

前記第二の終了メッセージを入力した場合には、第三の開始メッセージをパッシングする第三のパッシングステップと、

前記第三の開始メッセージを受け取った場合に、前記変換処理データについて出力処理を行い、出力処理が完了した場合には、第三の終了メッセージを出力する出力処理ステップと、

を含むことを特徴とする情報処理方法。

【請求項 6】 オブジェクト指向技術を応用したクラスを用いて情報処理を行う方法であって、

情報処理を行う内容が定義されたパラメータを入力する入力ステップと、

第一の開始メッセージをパッシングする第一のパッシングステップと、

前記第一の開始メッセージを受け取った場合に、パラメータに従って入力データを入力し、入力処理が完了した場合には、第一の終了メッセージを出力する入力処理ステップと、

前記第一の終了メッセージを入力した場合には、第二の開始メッセージをパッシングする第二のパッシングステップと、

前記第二の開始メッセージを受け取った場合に、パラメータに従って前記入力データの変換を行い、変換処理が完了した場合には、第二の終了メッセージを出力する変換処理ステップと、

前記第二の終了メッセージを入力した場合には、第三の開始メッセージをパッシングする第三のパッシングステップと、

前記第三の開始メッセージを受け取った場合に、前記変換処理データを入力し、パラメータに従って出力処理を行い、出力処理が完了した場合には、第三の終了メッセージを出力する出力処理ステップと、

を含むことを特徴とする情報処理方法。

【請求項 7】 前記入力ステップは、

ウィザード画面を用いて情報処理を行う内容が定義されたパラメータを入力することを特徴とする請求項 6 記載の情報処理方法。

【請求項 8】 オブジェクト指向技術を応用したクラス

を用いて情報処理を行うプログラムを記録したコンピュータ読み取り可能な記録媒体であって、
 第一の開始メッセージをパッシングする第一のパッシングステップと、
 前記第一の開始メッセージを受け取った場合に、データの入力を行って入力処理データを作成し、入力処理が完了した場合には、第一の終了メッセージを出力する入力処理ステップと、
 前記第一の終了メッセージを入力した場合には、第二の開始メッセージをパッシングする第二のパッシングステップと、
 前記第二の開始メッセージを受け取った場合に、前記入力データの変換を行って変換処理データを作成し、変換処理が完了した場合には、第二の終了メッセージを出力する変換処理ステップと、
 前記第二の終了メッセージを入力した場合には、第三の開始メッセージをパッシングする第三のパッシングステップと、前記第三の開始メッセージを受け取った場合に、前記変換処理データについて出力処理を行い、出力処理が完了した場合には、第三の終了メッセージを出力する出力処理ステップと、
 を含むことを特徴とする情報処理プログラムを記録したコンピュータ読み取り可能な記録媒体。

【請求項 9】 オブジェクト指向技術を応用したクラスを用いて情報処理を行うプログラムを記録したコンピュータ読み取り可能な記録媒体であって、
 情報処理を行う内容が定義されたパラメータを入力する入力ステップと、
 第一の開始メッセージをパッシングする第一のパッシングステップと、
 前記第一の開始メッセージを受け取った場合に、パラメータに従って入力データを入力し、入力処理が完了した場合には、第一の終了メッセージを出力する入力処理ステップと、
 前記第一の終了メッセージを入力した場合には、第二の開始メッセージをパッシングする第二のパッシングステップと、
 前記第二の開始メッセージを受け取った場合に、パラメータに従って前記入力データの変換を行い、変換処理が完了した場合には、第二の終了メッセージを出力する変換処理ステップと、
 前記第二の終了メッセージを入力した場合には、第三の開始メッセージをパッシングする第三のパッシングステップと、
 前記第三の開始メッセージを受け取った場合に、前記変換処理データを入力し、パラメータに従って出力処理を行い、出力処理が完了した場合には、第三の終了メッセージを出力する出力処理ステップと、
 を含むことを特徴とする情報処理プログラムを記録した

コンピュータ読み取り可能な記録媒体。

【請求項 10】 前記入カステップは、ウィザード画面を用いて情報処理を行う内容が定義されたパラメータを入力することを特徴とする請求項 9 記載の情報処理プログラムを記録したコンピュータ読み取り可能な記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、情報処理装置、情報処理方法及び情報処理プログラムを記録したコンピュータ読み取り可能な記録媒体に関し、特に、オブジェクト指向技術を応用したクラスを用いることにより、処理の手続きの内容等の変更を容易に行うことができ、さらに、プログラミングを必要とせずに情報処理を行うことが出来る技術に関する。

【0002】

【従来の技術】従来からコンピュータシステムを用いて情報処理を行う場合には、CやCOBOL等の高級言語等で記述することによりその処理の内容を特定していた。図16はソート処理を行うプログラム記述例である。このプログラムは、COBOLで記述しており、学生の成績結果が記録されている成績ファイル(SEISEKI-F)を読み込み、成績順位ファイル(JUNI-FILE)を作成する。成績ファイルのレコードは整列されておらず、この成績ファイルから合計点を降順に整列し、同得点の場合には、学籍番号の昇順に整列させる。また、成績順位ファイルのレコードには得点順位を得点の高いほうから順に付ける。このとき、同得点の場合には同順位とするようにしてある。

【0003】このようにして、従来から高級言語等で記述することにより情報処理を行っていた。

【0004】

【発明が解決しようとする課題】しかしながら、従来のCやCOBOL等の高級言語等で処理の内容を特定したのでは、入力ファイルが変更したり、手続きの内容が変更した場合には、プログラム開発者はプログラムを修正する必要がある。さらに、関連するプログラムを修正する必要がある。例えば、図16に示したプログラムでは、プログラム前半のファイル構造の定義等の変更が必要である。また、プログラム後半のメイン部では、ソートする際に昇順でソートか、降順でソートか等の手続きの変更や、入出力ファイルの変更等が必要である。今日のような複雑高度化したソフトウェア開発では時間的、人的に膨大な資源を要する。

【0005】本発明は上記事情に鑑みて成されたものであり、その目的は、処理の手続きの内容等の変更を容易に行うことができ、さらに、プログラミングを必要とせずに情報処理を行うことが出来る情報処理装置、情報処理プログラムの作成方法及び情報処理プログラムを作成するプログラムを記録したコンピュータ読み取り可能な

記録媒体を提供することにある。

【0006】

【課題を解決するための手段】本発明の発明者は、オブジェクト指向技術を応用したクラスを使ってソフトウェアを部品化し、その部品を組み合わせることにより、従来のプログラミングの手間を省くことが可能であることに気が付いた。オブジェクト指向の手順に基づく場合に、ソフトウェアシステムの分割単位にクラスなる概念を用いる。開発対象のソフトウェアシステムの構造を複数のクラスの集合体として取り扱う。各クラスの特徴とクラス間の関係は、オブジェクト指向データモデルが利用される。

【0007】オブジェクト指向データモデルに属する各クラスには、メンバとメソッドの要素があり、クラスは少なくとも一つのメンバを有している。各クラスは、パラメータにより振る舞いが定義される。パラメータは、オブジェクト指向データモデルの振る舞いを定義するためのインスタンス（インスタンス値）をプログラム内に埋め込むことにより処理が行われる。従って、ここでいうパラメータには、振る舞いを定義するインスタンス（インスタンス値）が含まれる。また、メソッドとは、各クラスによって実行される手続きを指す。各クラス間は、メソッドにおいて定義された応答であるメッセージの送受（パッシング）が行われる。メッセージには、操作依頼や、操作結果の授受が含まれる。

【0008】このような、オブジェクト指向技術を応用しつつ慎重な研究を重ねた結果、発明者は、以下の発明を行うことが出来た。

【0009】請求項1の発明は、オブジェクト指向技術を応用したクラスを用いて情報処理を行う装置であって、開始メッセージをパッシングし、クラスからの終了メッセージの入力により次のクラスをスケジュールするバッチコントロールクラスと、前記バッチコントロールクラスからの開始メッセージを受け取った場合に、所定の条件でデータの入力処理を行い、前記バッチコントロールクラスへ終了メッセージを出力する入力クラスと、前記バッチコントロールクラスからの開始メッセージを受け取った場合に、所定の条件で前記入力処理を行ったデータの変換を行い、前記バッチコントロールクラスへ終了メッセージを出力する変換クラスと、前記バッチコントロールクラスからの開始メッセージを受け取った場合に、所定の条件で前記変換処理データについて出力処理を行い、前記バッチコントロールクラスへ終了メッセージを出力する出力クラスと、を備えることを特徴とする。

【0010】請求項2の発明は、前記請求項1における所定の条件は、各クラスの処理内容を定義するパラメータにより特定されることを特徴とする。

【0011】上記請求項1及び2の発明の構成では、オブジェクト指向技術を応用したクラスを用いてソフトウ

エアを部品化する。部品は、入力クラス、変換クラス、出力クラス、及びこれらクラスの管理を行うバッチコントロールクラスを用いる。入力クラス、変換クラス、及び出力クラスのパラメータを定義することにより、各クラスの処理内容が決定する。このようにすることで、処理の手続きの内容等の変更を容易に行うことができるようになるのである。

【0012】請求項3の発明は、オブジェクト指向技術を応用したクラスを用いて情報処理を行うプログラムを作成する装置であって、各クラスの処理内容を定義するパラメータを入力する入力部と、入力した前記パラメータに基づいて、開始メッセージをパッシングし、クラスからの終了メッセージの入力により次のクラスをスケジュールするバッチコントロールクラス、前記バッチコントロールクラスからの開始メッセージを受け取った場合に、所定の条件でデータの入力処理を行い、前記バッチコントロールクラスへ終了メッセージを出力する入力クラス、前記バッチコントロールクラスからの開始メッセージを受け取った場合に、所定の条件で前記入力処理を行ったデータの変換を行い、前記バッチコントロールクラスへ終了メッセージを出力する変換クラス、及び、前記バッチコントロールクラスからの開始メッセージを受け取った場合に、所定の条件で前記変換処理データについて出力処理を行い、前記バッチコントロールクラスへ終了メッセージを出力する出力クラス、を実行するプログラムを生成するプログラム生成部と、を備えることを特徴とする。

【0013】また、請求項4の発明は、前記請求項3の入力部は、ウィザード画面を用いて入力を受け付けることを特徴とする。

【0014】請求項3及び4の発明によれば、処理の手続きの内容等の変更を容易に行うことができるプログラムを容易に生成することができ、さらに、プログラミングを必要とせずにプログラムを生成することができる。

【0015】上記目的を達成するため、請求項5の発明は、オブジェクト指向技術を応用したクラスを用いて情報処理を行う方法であって、第一の開始メッセージをパッシングする第一のパッシングステップと、前記第一の開始メッセージを受け取った場合に、データの入力を行う入力処理データを作成し、入力処理が完了した場合には、第一の終了メッセージを出力する入力処理ステップと、前記第一の終了メッセージを入力した場合には、第二の開始メッセージをパッシングする第二のパッシングステップと、前記第二の開始メッセージを受け取った場合に、前記入力データの変換を行って変換処理データを作成し、変換処理が完了した場合には、第二の終了メッセージを出力する変換処理ステップと、前記第二の終了メッセージを入力した場合には、第三の開始メッセージをパッシングする第三のパッシングステップと、前記第三の開始メッセージを受け取った場合に、前記変換処

理データについて出力処理を行い、出力処理が完了した場合には、第三の終了メッセージを出力する出力処理ステップと、を含むことを特徴とする。

【0016】請求項6の発明は、オブジェクト指向技術を応用したクラスを用いて情報処理を行う方法であって、情報処理を行う内容が定義されたパラメータを入力する入力ステップと、第一の開始メッセージをパッシングする第一のパッシングステップと、前記第一の開始メッセージを受け取った場合に、パラメータに従って入力データを入力し、入力処理が完了した場合には、第一の終了メッセージを出力する入力処理ステップと、前記第一の終了メッセージを入力した場合には、第二の開始メッセージをパッシングする第二のパッシングステップと、前記第二の開始メッセージを受け取った場合に、パラメータに従って前記入力データの変換を行い、変換処理が完了した場合には、第二の終了メッセージを出力する変換処理ステップと、前記第二の終了メッセージを入力した場合には、第三の開始メッセージをパッシングする第三のパッシングステップと、前記第三の開始メッセージを受け取った場合に、前記変換処理データを入力し、パラメータに従って出力処理を行い、出力処理が完了した場合には、第三の終了メッセージを出力する出力処理ステップと、を含むことを特徴とする。

【0017】請求項7の発明は、前記請求項6における入力ステップは、ウィザード画面を用いて情報処理を行う内容が定義されたパラメータを入力することを特徴とする。

【0018】上記目的を達成するため、請求項8の発明は、オブジェクト指向技術を応用したクラスを用いて情報処理を行うプログラムを記録したコンピュータ読み取り可能な記録媒体であって、第一の開始メッセージをパッシングする第一のパッシングステップと、前記第一の開始メッセージを受け取った場合に、データの入力を行って入力処理データを作成し、入力処理が完了した場合には、第一の終了メッセージを出力する入力処理ステップと、前記第一の終了メッセージを入力した場合には、第二の開始メッセージをパッシングする第二のパッシングステップと、前記第二の開始メッセージを受け取った場合に、前記入力データの変換を行って変換処理データを作成し、変換処理が完了した場合には、第二の終了メッセージを出力する変換処理ステップと、前記第二の終了メッセージを入力した場合には、第三の開始メッセージをパッシングする第三のパッシングステップと、前記第三の開始メッセージを受け取った場合に、前記変換処理データについて出力処理を行い、出力処理が完了した場合には、第三の終了メッセージを出力する出力処理ステップと、を含むことを特徴とする。

【0019】請求項9の発明は、オブジェクト指向技術を応用したクラスを用いて情報処理を行うプログラムを記録したコンピュータ読み取り可能な記録媒体であつ

て、情報処理を行う内容が定義されたパラメータを入力する入力ステップと、第一の開始メッセージをパッシングする第一のパッシングステップと、前記第一の開始メッセージを受け取った場合に、パラメータに従って入力データを入力し、入力処理が完了した場合には、第一の終了メッセージを出力する入力処理ステップと、前記第一の終了メッセージを入力した場合には、第二の開始メッセージをパッシングする第二のパッシングステップと、前記第二の開始メッセージを受け取った場合に、パラメータに従って前記入力データの変換を行い、変換処理が完了した場合には、第二の終了メッセージを出力する変換処理ステップと、前記第二の終了メッセージを入力した場合には、第三の開始メッセージをパッシングする第三のパッシングステップと、前記第三の開始メッセージを受け取った場合に、前記変換処理データを入力し、パラメータに従って出力処理を行い、出力処理が完了した場合には、第三の終了メッセージを出力する出力処理ステップと、を含むことを特徴とする。

【0020】請求項10の発明は、前記請求項9における入力ステップは、ウィザード画面を用いて情報処理を行う内容が定義されたパラメータを入力することを特徴とする。

【0021】

【発明の実施の形態】本発明に係る情報処理装置、情報処理プログラムの作成方法及び情報処理プログラムを作成するプログラムを記録したコンピュータ読み取り可能な記録媒体の実施形態について、図面を参照しながら詳細に説明する。

【0022】本発明に係る情報処理装置が具備するハードウェア構成は、各種処理を行うためのCPUと、キーボード、マウス、ライトペン、又はフレキシブルディスク装置等の入力装置と、メモリ装置やディスク装置等の外部記憶装置と、ディスプレイ装置、プリンタ装置等の出力装置等とを備えた通常のコンピュータシステムを用いる。このコンピュータシステムには、いわゆる汎用機、ワークステーション、及びパーソナルコンピュータが含まれる。なお、前記CPUは、後述する各クラスにおける処理等を行う演算部と、前記処理の命令や各種のデータ等を記憶する主記憶部とを具備する。

【0023】図1は、本実施形態の情報処理装置を示すブロック図である。

【0024】本実施形態の情報処理装置は、スケジュールのパラメータに従い、開始メッセージをパッシングし、クラスからの終了メッセージの入力により次のクラスをスケジュールするバッチコントロールクラス11と、前記バッチコントロールクラス11からの開始メッセージを受け取った場合に、パラメータに従って入力データを入力し、入力処理が完了した場合には、前記バッチコントロールクラス11へ終了メッセージを出力する入力クラス13と、前記バッチコントロールクラス11

からの開始メッセージを受け取った場合に、パラメータに従って前記入力データの変換を行い変換処理が完了した場合には、前記バッチコントロールクラス11へ終了メッセージを出力する変換クラス15と、前記バッチコントロールクラス11からの開始メッセージを受け取った場合に、前記変換処理データを入力し、パラメータに従って出力処理を行い、出力処理が完了した場合には、前記バッチコントロールクラス11へ終了メッセージを出力する出力クラス17と、を備える。

【0025】ここで、バッチコントロールクラス11は、総てのバッチクラス（入力クラス、変換クラス、及び出力クラス）をコントロールする。バッチクラスのコントロールは、スケジュールのパラメータ（典型的にはインスタンス値）に従って、各クラスに開始メッセージ（典型的にはイベント）をパッシングし、終了メッセージ（典型的にはイベント）を受け取り次のクラスをスケジュールする。クラスのスケジュールは、入力クラス、変換クラス、出力クラスの順で行われる。すなわち、まず、入力クラスに開始メッセージをパッシングし、入力クラスから終了メッセージを受け取った場合には、次に、変換クラスに開始メッセージをパッシングする。変換クラスから終了メッセージを受け取った場合には、次に出力クラスに開始メッセージをパッシングする。

【0026】スケジュールのパラメータは、処理の内容を記述するバッチプログラムを作成する際に決定される。クラスライブラリから処理に必要なプログラムソースコードを入力し、そのプログラムソースコードにパラメータをインスタンス値として埋め込むことでプログラムが完成する。また、スケジュールのパラメータは、バッチ情報インスタンス保持部23に保持するようにして、必要に応じてバッチコントロールクラス11が参照する。

【0027】入力クラス13は、バッチコントロールクラス11からの開始メッセージを受け取った場合に、処理の手順や内容が示されたパラメータに従って入力データを入力データ保持部25から入力し、入力処理が完了した場合には、バッチコントロールクラス11へ終了メッセージを出力する。入力処理は、入力全体処理用、参照用のインデックスファイルを作成したり、入力ファイルのフォーマット変換、2つのファイルのマッチングやソートが含まれる。

【0028】変換クラス15は、バッチコントロールクラス11からの開始メッセージを受け取った場合に、処理の手順や内容が示されたパラメータに従って入力データの変換を行い変換処理が完了した場合には、前記バッチコントロールクラス11へ終了メッセージを出力する。この変換クラスは、まず、コントロールクラスからの開始メッセージによって起動される。パラメータによって定義された各種のサービスメソッドにより処理を行う。処理結果のステータス（例外、中断、正常終了（処

理件数等）のメッセージをジョブコントロールクラス19にパッシングする。処理が終了した場合にはバッチコントロールクラス11に終了メッセージを出力する。

【0029】ここで、主なメソッドは大きく分けて、変換メソッド、チェックメソッド、サービスメソッドがあり、それぞれのメソッドを1回若しくは複数回繰り返すことにより、データの処理を施す。処理が終了したデータは変換ワークとして出力される。

【0030】出力クラス17は、変換クラス15によって処理が施された変換処理データ（変換ワーク）について、指定された出力媒体に出力するクラスである。この出力クラス17は、バッチコントロールクラス11からの開始メッセージを受け取った場合に、前記変換処理データを入力し、処理の手順や内容が示されたパラメータに従って出力処理を行い、出力処理が完了した場合には、前記バッチコントロールクラス11へ終了メッセージを出力する。操作者は、形式によって、例えば、テキスト形式、CSV形式、インポート形式、DB出力、印書出力等の形式を指定して出力させることが出来る。出力したデータは出力データ保持部27に保持され、またはプリンタに印字される。

【0031】ジョブコントロールクラス19は、変換クラス15からのステータスメッセージを受け取って、ログファイル29にステータスを書き込んだり、ジョブ管理システム21にステータス（処理件数等）を出力する。

【0032】ジョブ管理システム21は、ジョブコントロールシステム21からの処理件数等のステータスを入力してジョブの管理を行う。

【0033】図2は、本実施形態のクラスダイアグラムである。

【0034】ここで、バッチコントロールクラスのスケジュールデータは、各バッチクラスの処理の流れを定義したものである。この定義はパラメータ設定により行われる。

【0035】入力クラスのメソッドには、入力ファイルをソートしつつ入力するソート入力メソッド、複数の入力ファイルをマージしつつ入力するマージ入力メソッド、複数の入力ファイルのマッチング処理を行いつつ入力するマッチング入力メソッド、データベースから入力するDB入力メソッド、及びファイルから入力するファイル入力メソッドが含まれる。

【0036】また、変換クラスのメソッドには、ファイル内のデータのソートを行うソートメソッド、ファイル内のデータのフォーマット変換を行うフォーマットメソッド、ファイル内のデータの変換を行う置換メソッド、ファイル内のデータの展開を行う展開メソッド、複数のファイルのマッチング処理を行うマッチングメソッド、及びファイル内のデータの集計を行う集計サマリメソッドが含まれる。処理が施されたデータは、変換ワークと

して出力される。

【0037】また、出力クラスの方法には、変換処理ファイルをテキスト形式で出力を行うTXTメソッド、変換処理ファイルをCSV形式で出力を行うCSVファイル出力メソッド、変換処理ファイルをインポート形式で出力を行うインポート形式ファイル出力メソッド、変換処理ファイルを所定のアカウントへメールするメール出力メソッド、変換処理ファイルを帳票へ出力する帳票出力メソッド、変換処理ファイルをデータベースへ出力するDB出力メソッド、及び変換処理ファイルを印書へ出力する印書出力メソッドが含まれる。

【0038】図3は、変換クラスの変換処理を説明するための図である。

【0039】まず、フォーマットメソッドのフォーマット変換は、ファイル内のデータのフォーマット変換を行う。例えば、ファイル内のデータが97/12/01なる形式で記述されている場合に、1997年12月1日なる形式に変換したり、ファイル内のデータが1000なる形式で記述されている場合に、¥1,000なる形式に変換したりする如くである。

【0040】また、置換メソッドは、ファイル内のデータの置換処理を行う。例えば、ファイル内のデータが100EAなる形式で記述されている場合に、100個なる形式に置換したりする如くである。

【0041】また、展開メソッドは、ファイル内のデータの展開処理を行う。例えば、ファイル内のデータが親品番/子品番というようにデータがスラッシュ(/)で分けられて記述されている場合に、図示の如く、そのデータの行を替えて記述したりする如くである。

【0042】図4は、本実施形態の情報処理装置が実行する実行ファイルの作成手順を示すブロック図である。まず、既存のクラスライブラリ31から処理に必要なクラスをコピーする。続いて、ウィザードを用いたインスタンス値の定義部33にて、コピーしたクラスのインスタンス値をウィザードを用いてインスタンス値の定義を行い、インスタンス値が埋め込まれたプログラムソースコード35を作成する。これについては後に詳述する。このプログラムソースコード35はコンパイラ37にてコンパイル処理がされ、コンピュータにて実行可能な実行ファイル39が作成される。この実行ファイル39を実行することにより、本実施形態の情報処理が実現する。

【0043】図5乃至15は、ウィザードを用いたインスタンス値の定義部33の処理を示す画面例である。これらの画面はグラフィクスユーザインターフェース(GUI)を実現しており、キーボードによる入力、マウス等を用いて入力を行う。

【0044】図5は、本実施形態のバッチ定義画面である。

【0045】まず、クラスライブラリからソースファイ

ルを特定する。ここで、“参照”ボタンをクリックするとクラスライブラリの一覧が表示されるようにしてある。また、変換ソースファイルは、入力するソースファイルと異なったファイル名を付ける場合には、新たなソースファイル名を指定する。

【0046】次に、入力情報として、入力ファイルの定義を行う。複数の入力ファイルの場合があるので、各入力ファイル毎に指定できるようにしてある。本実施形態では、2つのファイルについて示したが、これに限られることはない。定義が終了した場合には“入力ファイル登録”ボタンをクリックすると更に詳細な定義を行う画面(図6)を表示する。

【0047】次に、変換情報として、変換処理の定義を行う。ここでは、行うべき処理(メソッド)やその順序の定義を行う。変換処理の種類はリストとして表示され、各変換処理のボタンをクリックして“追加”ボタンをクリックすると処理順序に追加される。そこで、“移動”ボタン等をクリックすることにより順序を変更することが出来る。“編集”ボタンをクリックすると詳細な定義を行う画面(図7乃至図12)を表示する。

【0048】次に、出力情報として出力ファイルの定義を行う。複数の出力を行う場合があるので、各出力の順序等を定義できるようにしてある。定義が終了した場合には“編集”ボタンをクリックすると、各出力方法について更に詳細な定義を行う画面(図14)を表示する。

【0049】総ての定義を終了した場合には、“OK”ボタンをクリックする。これにより、新たなプログラムソースコードが生成される。図4で説明したとおり、このプログラムソースコードをコンパイラを用いて実行ファイルを作成する。

【0050】図6は、本実施形態の入力ファイルの詳細な定義を行う画面である。まず、その入力ファイルの形式を定義する。本実施形態の場合には、テキスト形式かDB形式かを入力するようにしてある。“OK”ボタンをクリックすると、テキスト形式を指定した場合には、図面右上の画面を表示する。ここでは、入力ファイルの区切り文字の情報や出力ファイル名を指定する。さらに、その出力ファイルをソート、マージ等の処理を行う場合には、その処理を定義する。ソート処理の場合には、どのコラムでソートするか、また、昇順か降順の別を定義する。複数のキーによりソートする場合には、そのキーとそのソート順を定義する。

【0051】また、DB形式を指定した場合には、図面右下の画面を表示する。ここでは、DB名、テーブル名、列名、WHERE句等を定義する。両画面とも“OK”画面により登録される。

【0052】図7は、本実施形態の変換処理のうち、部分置換の詳細な定義を行う画面である。まず、部分置換を行うファイル等を指定する。一般には、入力ファイル定義にて指定された出力ファイルがデフォルトで定義

される。但し、変換処理が複数行われる場合には、図 5 に示した画面の変換順序で定義された前の処理の出力ファイルがデフォルトで定義される。置換情報の定義は、コード置換、コード分離、及びコード結合等、必要な情報を入力することにより指定を行う。総ての定義を終了した場合には、“OK” ボタンをクリックする。

【0053】図 8 は、本実施形態の変換処理のうち、全体置換の詳細な定義を行う画面である。まず、全体置換を行うファイル等を指定する。一般には、入力ファイル定義にて指定された出力ファイルがデフォルトで定義される。但し、変換処理が複数行われる場合には、図 5 に示した画面の変換順序で定義された前の処理の出力ファイルをデフォルトで定義される。置換情報の定義は、文字列の挿入、特定文字の挿入、及びスペースコンパクション等、必要な情報を入力することにより指定を行う。総ての定義を終了した場合には、“OK” ボタンをクリックする。

【0054】図 9 は、本実施形態の変換処理のうち、条件置換の詳細な定義を行う画面である。まず、条件置換を行うファイル等を指定する。一般には、入力ファイル定義にて指定された出力ファイルがデフォルトで定義される。但し、変換処理が複数行われる場合には、図 5 に示した画面の変換順序で定義された前の処理の出力ファイルをデフォルトで定義される。置換情報の定義は、条件置換のロジックについての必要な情報を入力することにより指定を行う。総ての定義を終了した場合には、“OK” ボタンをクリックする。

【0055】図 10 は、本実施形態の変換処理のうち、条件置換の詳細な定義を行う画面である。まず、入力データを指定し、対象項目として集計を行うカラムの指定を行う。次に、データ項目名を定義してその計算式を関数メソッド、演算子、数値パッド等を用いて定義する。総ての定義を終了した場合には、“OK” ボタンをクリックする。

【0056】図 11 (a) は、本実施形態のマッチング処理の詳細な定義を行う画面である。まず、マッチングを行うキーを開始カラム及び終了カラムにて入力する。次に、これらのキーの条件を入力して、処理相対番号を入力する。また、入力項目及び出力項目そして呼出メソッドを入力する。総ての定義を終了した場合には、“OK” ボタンをクリックする。

【0057】図 11 (b) は、マッチング処理の例である。例えばファイル 1 に A, B, C からなるデータが格納されており、また、ファイル 2 に A, D, E が格納されているものとする。この場合には、マッチング処理により、ファイル 3 として A, B, D, C, E からなるデータが格納されることとなる。総ての定義を終了した場合には、“OK” ボタンをクリックする。

【0058】図 12 は本実施形態のチェック処理の詳細な定義を行う画面である。まず、入力データを定義し

て、入力データ中の対象カラムを定義する。また、チェック結果の出力としてメッセージダイアログ形式で出力するか、テキストとして出力するか、印書として出力するか等を定義する。チェックのメソッドが予めある場合には、そのチェックメソッドを定義する。ここで、フォーマットチェックの場合は、対象項目が、YY/MM/DD というフォーマットであるか否か等を指定する。この指定されたフォーマット以外の場合には、出力形式で指定された形式で出力される。総ての定義を終了した場合には、“OK” ボタンをクリックする。

【0059】図 13 は、本実施形態のステータスマニタの定義画面である。この画面は、各クラスの処理状況をステータスとしてモニタリングするものである。入力処理のステータスとしては、開始時間、進捗、終了時間等が表示される。また、変換処理のステータスとしては、進捗等が表示される。出力処理のステータスとしては、開始時間、進捗、終了時間等が表示される。操作者は、これらの情報により、処理の状況を把握することが出来るのである。

【0060】図 14 は、本実施形態の出力ファイルの詳細な定義を行う画面である。まず、出力するファイルを指定する。次に出力先として、その出力ファイルの形式を定義する。本実施形態の場合には、テキスト形式、CSV 形式、DB 形式、プリンタを定義するようにしてある。テキスト形式の場合には、出力先のファイルを定義する。CSV 形式の場合には、出力先のファイル及びカラムを定義する。また、DB 形式の場合には、DB 名、テーブル名等を定義する。プリンタ出力の場合には、出力するカラム等を定義する。総ての定義を終了した場合には、“OK” ボタンをクリックする。

【0061】図 15 は、本実施形態のジョブコントロールクラス 19 についてのインスタンス値を定義する画面である。この画面は、変換クラスにおける各処理について、開始/終了メッセージ、アクセス件数、ワーニング件数等の必要な情報を定義する。またログの出力先はファイル形式か DB 形式かの定義を行い、ファイル名や DB 名等を定義する。定義された情報は、ログとしてログ保持部 29 に出力される。

【0062】以上のような画面を用いて、操作者に入力クラス、バッチジョブコントロールクラス、及び出力クラスのメソッド及びインスタンス値を入力させ、処理が特定されたプログラムソースコードが作成される。そのプログラムソースコードをコンパイルしてコンピュータ実行可能な実行ファイルを作成し、その実行ファイルをコンピュータが実行することで、本実施形態を実施することが出来る。

【0063】なお、上述したウィザード定義部やそれにより生成されたプログラムコード及び実行ファイルは記録媒体に保存することができる。この記録媒体をコンピュータシステムによって読み込ませ、前記プログラムを

実行してコンピュータを制御しながら上述した情報処理を実現することができる。ここで、前記記録媒体とは、メモリ装置、磁気ディスク装置、光ディスク装置等、プログラムを記録することができるような装置が含まれる。

【0064】

【発明の効果】以上説明したように本発明によれば、作成若しくは生成したプログラムの変更を容易に行うことが出来るため、再利用性が高く、保守性が高いシステムを提供することが出来る。これにより、生産性の高いシステムを提供することができる。

【図面の簡単な説明】

【図1】本実施形態の情報処理装置を示すブロック図である。

【図2】本実施形態のクラスダイアグラムである。

【図3】変換クラスの変換処理を説明するための図である。

【図4】本実施形態の情報処理装置が実行する実行ファイルの作成手順を示すブロック図である。

【図5】本実施形態のバッチ定義画面である

【図6】本実施形態の入力ファイルの詳細な定義を行う画面である。

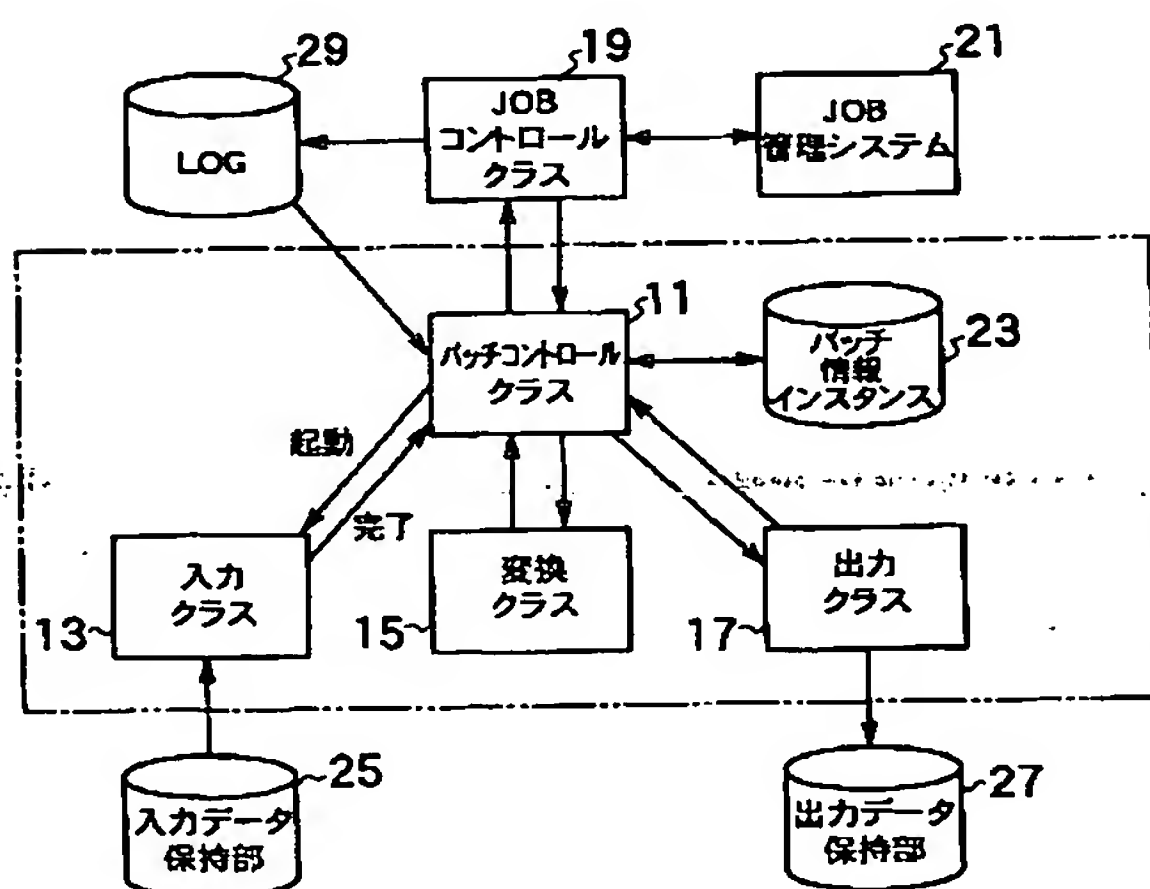
【図7】本実施形態の変換処理のうち、部分置換の詳細な定義を行う画面である。

【図8】本実施形態の変換処理のうち、全体置換の詳細な定義を行う画面である。

【図9】本実施形態の変換処理のうち、条件置換の詳細な定義を行う画面である。

【図10】本実施形態の集計サマリの詳細な定義を行う画面である。

【図1】



【図11】(a)は、本実施形態のマッチング処理の詳細な定義を行う画面であり、(b)は、マッチング処理の例である。

【図12】本実施形態のチェック処理の詳細な定義を行う画面である。

【図13】本実施形態のステータスマニタの定義画面である。

【図14】本実施形態の出力ファイルの詳細な定義を行う画面である。

【図15】本実施形態のジョブコントロールクラス19についてのインスタンス値を定義する画面である。

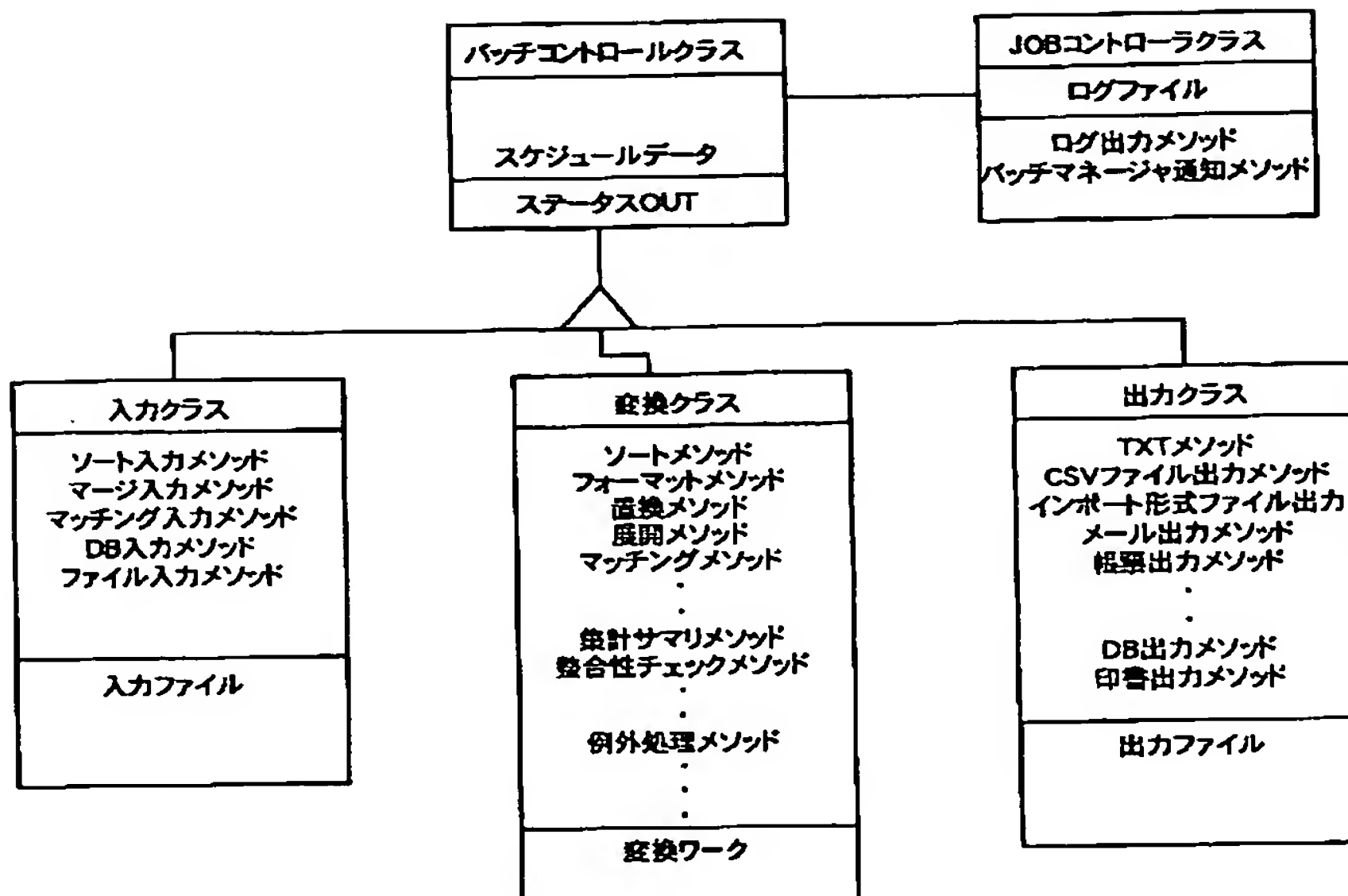
【図16】COBOLを用いて記述したソートプログラムの例である。

【符号の説明】

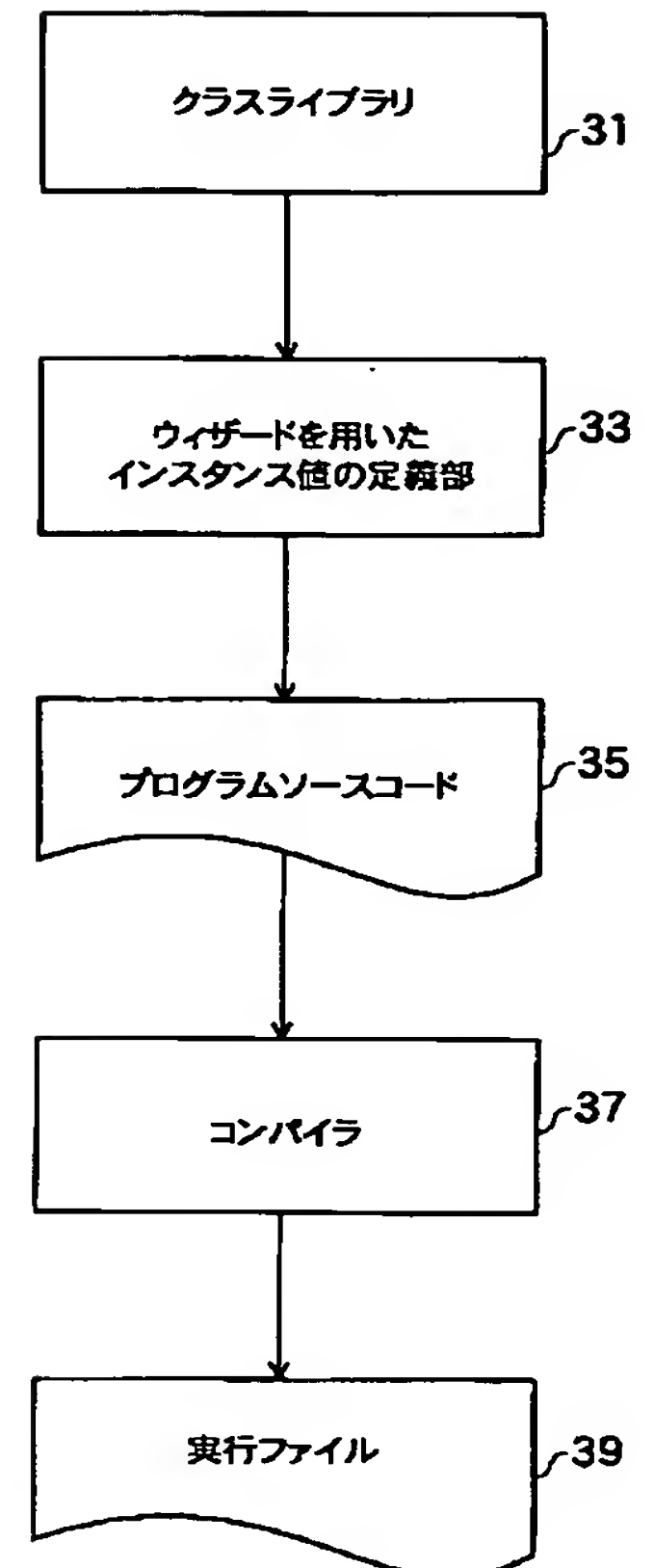
- 11 バッチコントロールクラス
- 13 入力クラス
- 15 変換クラス
- 17 出力クラス
- 19 ジョブコントロールクラス
- 21 ジョブ管理システム
- 23 バッチ情報インスタンス
- 25 入力データ保持部
- 27 出力データ保持部
- 29 ログ保持部
- 31 クラスライブラリ
- 33 ウィザードを用いたインスタンス値の定義部
- 35 プログラムソースコード
- 37 コンパイラ
- 39 実行ファイル

【図10】

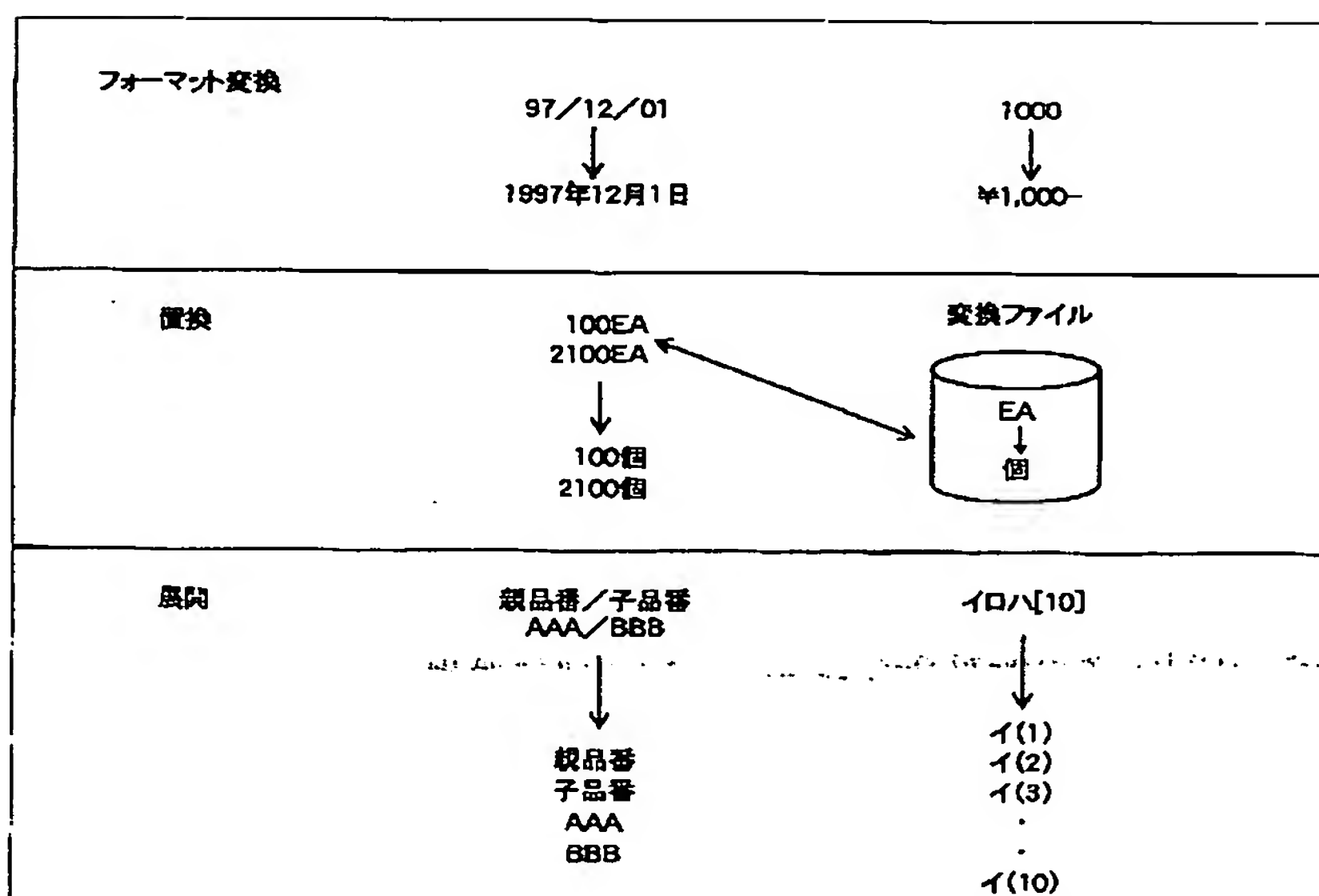
【図 2】



【図 4】



【図 3】



【図5】

ソースファイル名 参照

変換ソースファイル 参照

入力情報

☒ 入力ファイル1 ☐ 入力ファイル2

入力ファイル定義 参照 C:\FORTEN\ACT\WINTERFACE.DAT

データイメージ 1997/03 00-0000-0001 00-0000-0002

変換情報

☐ 全体置換 ☒ 部分置換 ☐ 条件置換

☐ 全体結合 ☐ 部分結合

☐ ソート ☐ マージ ☐ マッチ

☐ チェック ☐ サービス

1 ソート1
 2 部分置換1
 3 ソート2
 4 全体置換1

出力情報

出力ファイル定義 参照 C:\FORTEN\ACT\WINTERFACE.DAT

データイメージ 1997/03 00-0000-0001 00-0000-0002

1 テキスト出力1
 2 DB出力1
 3 印字1
 4 DB出力2

【図6】

入力ファイル定義

☒ テキスト ☐ DB

入力ファイル定義(テキスト)

☒ ファイル1 ☐ ファイル2 入力ファイル名

区別文字 区別文字()

データイメージ

出力ファイル名 参照

☐ ソート ☐ マージ

ソートキー

開始位置 桁数

☒ 昇順 ☐ 降順

1 12-3 A
 2 15-8 D
 3 20-8 A

入力ファイル定義(DB)

DB名

テーブル名

列名

照会条件

データイメージ

出力ファイル名 参照

1 COMPANY-ID
 2 COMPANY-NAME
 3 ZIP-CODE

【図7】

入力ファイル 出力ファイル
 入力イメージ 出力イメージ
 変換名称
 入力ファイル 開始位置 桁数

変換情報

☒ コード変換
☐ メモリ展開

ファイル名等
 キー 開始位置 桁数
 変換 開始位置 桁数

DB DB名 ログイン
 テーブル名
 列名
 キー 開始位置 桁数
 変換 開始位置 桁数
 条件可

☐ コード分離 分離文字
 要素数 (カンマで区切る)
☐ コード結合 条件文字

変換名	変換情報
コード変換	15-8/12-4

【図8】

入力ファイル 出力ファイル
 入力イメージ 出力イメージ

変換情報

☐ 文字列挿入 挿入文字列 開始位置
☐ 特定文字置換 ABC → 2 2 開始位置
☐ スペースコンパクション 開始位置 桁数

変換名

【図9】

入力ファイル 出力ファイル
 入力イメージ 出力イメージ

条件置換

ロジック 1 開始位置 桁数
 ロジック 2

条件置換

【図11】

(a) **条件置換**

キー1 開始位置 桁数
 キー2 開始位置 桁数

判定条件

☐ キー1 = キー2 ☐ キー1 ≥ キー2
☐ キー1 > キー2 ☐ キー1 ≤ キー2
☐ キー1 < キー2

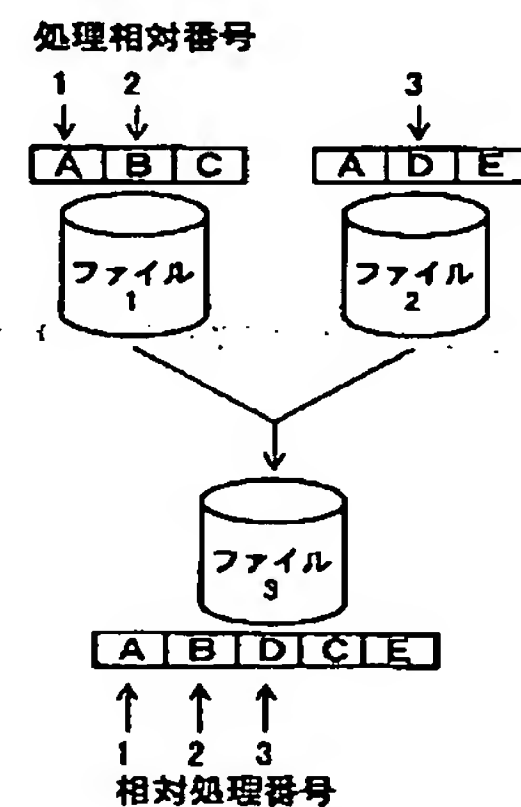
処理相対番号

入力項目 開始位置 桁数
 出力項目 開始位置 桁数
 呼び出しメソッド

入力ファイルイメージ1
 2

出力ファイルイメージ

(b)



【图 13】

[illegible]

【図 15】

サービス

システムコントロールログ

変換

1	SORT	▲
2	置換	▼

☐ 開始/終了メッセージ

ファイル

ファイルA ファイルB	▲ ▼
----------------	--------

☐ アクセス件数
☐ ワーニング件数
☐ スキップ件数
☐ 挿入件数
☐ 削除件数
☐ 更新件数
☐ 抽出件数
☐ ステータスメッセージ 件毎に出力

出力先

☒ ファイル

☐ DB8
 DB名

テーブル

【図 16】

```

DATA DIVISION
FD SEISEKI-F.
01 I-SEI-REC.
02 I-NO-NAME          PIC X(28).
02 I-TBL              PIC 9(3) OCCURS 5.
03 I-KAMOKU
SD SORT-FILE
01 SORT-REC.
02 S-NO-NAME          PIC X(8).
03 S-GAKU-NO          PIC X(20).
03 S-NAM3             PIC 9(3).
02 S-GOUKEI           PIC 9(3).
02 S-TOKUTEN          PIC 9(15).
FD JUNI-FILE
01 O-JUN-REC.
02 O-JUNI             PIC X(3).
02 O-NO-NAME          PIC X(28).
02 O-GOUKEI           PIC 9(3).
02 O-TOKUTEN          PIC 9(15).
WORKING-STORAGE SECTION.
01 I                  PIC 9          VALUE ZERO.
01 EOF                PIC 9.
01 W-JUNI1             PIC 9(3).
01 W-JUNI2             PIC 9(3).
01 W-GOUKEI           PIC 9(3).
PROCEDURE DIVISION.
MAIN.
  SORT SORT-FILE
  DESCENDING KEY S-GOUKEI
  ASCENDING KEY S-GAKU-NO.
  INPUT PROCEDURE IN-SHORI
  OUTPUT PROCEDURE OUT-SHORI
  STOP RUN.
IN-SHORI.
  OPEN INPUT SEISEKI-F.
  INITIALIZE EOF.
  PERFORM UNTIL EOF = 1
    READ SEISEKI-F
    AT END
      MOVE 1 TO EOF.
    NOT AT END
      MOVE I-NO-NAME TO S-NO-NAME
      MOVE I-TBL TO S-TOKUTEN
      PERFORM VARYING I FROM 1 BY 1 UNTIL I > 5
      COMPUTE S-GOUKEI = I-KAMOKU(I) + S-GOUKEI
    END-PERFORM
  END-READ
  RELEASE SORT-REC
  END-PERFORM.
CLOSE SEISEKI-F
OUT-SHORI.
  OPEN OUTPUT JUNI-FILE.
  INITIALIZE EOF W-JUNI1 W-JUNI2.
  MOVE HIGH-VALUE TO W-GOUKEI.
  PERFORM UNTIL EOF = 1
    RETURN SORT-FILE
    AT END
      MOVE 1 TO EOF
    NOT AT END
      COMPUTE W-JUNI1 = W-JUNI1 + 1
      IF W-GOUKEI NOT = S-GOUKEI
        COMPUTE W-JUNI2 = W-JUNI2 + W-JUNI1
      END-IF
      MOVE W-JUNI2 TO O-JUNI
      MOVE S-NO-NAME TO O-NO-NAME
      MOVE S-GOUKEI TO O-GOUKEI
      MOVE S-TOKUTEN TO O-TOKUTEN
      WRITE O-JUN-REC
      MOVE S-GOUKEI TO W-GOUKEI
    END-RETURN
  END-PERFORM.
CLOSE JUNI-FILE.

```


This Page is Inserted by IFW Indexing and Scanning Operations and is not part of the Official Record.

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

☒ **BLACK BORDERS**

☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**

☐ **FADED TEXT OR DRAWING**

☒ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**

☐ **SKEWED/SLANTED IMAGES**

☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**

☐ **GRAY SCALE DOCUMENTS**

☒ **LINES OR MARKS ON ORIGINAL DOCUMENT**

☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**

☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.